Received November 25, 2016, accepted December 24, 2016, date of publication January 4, 2017, date of current version January 23, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2647824

Secure Pervasive Social Communications Based on Trust in a Distributed Way

CHAOYIN HUANG[1], ZHENG YAN[1,2], (Senior Member, IEEE), NING LI[1], AND MINGJUN WANG[1]

[1]State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an 710071, China
[2]Department of Communications and Networking, Aalto University, Espoo 02150, Finland

Corresponding author: Z. Yan (zyan@xidian.edu.cn)

This work was supported in part by the National Key Foundational Research and Development on Network and Space Security, China, under Grant 2016YFB0800704, in part by the NSFC under Grant 61672410 and Grant U1536202, in part by the Project Supported by Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2016ZDJC-06, in part by the PhD grant of the Ministry of Education, China, under Grant JY0300130104, in part by the 111 project under Grant B08038 and Grant B16037, and in part by Aalto University.

ABSTRACT Social network has extended its popularity from the Internet to mobile domain. Pervasive social networking (PSN) supports instant social activities based on self-organized mobile ad hoc networks. PSN is useful in reality when fixed networks are unavailable or inconvenient to access or when people are in vicinity. For supporting crucial PSN activities and enhancing user privacy, securing pervasive social communications becomes important. However, a solution based on a centralized server could be inapplicable in some specific situations (e.g., disasters and military activities) and suffers from DoS/DDoS attacks and internal attacks. How to automatically control data access in a trustworthy and efficient way in PSN is a challenge. In this paper, we propose two schemes to secure communication data in PSN purely based on local trust evaluated by PSN nodes in a distributed manner. Each node can control its data based on its trust in other nodes by applying attribute-based encryption. The advantages, security, and performance of the proposed scheme are evaluated and justified through serious analysis and implementation. The results show the efficiency and effectiveness of the schemes. In addition, we developed a mobile app based on Android platform to demonstrate the applicability and social acceptance of our schemes.

INDEX TERMS Trust, social networking, data access control, attribute-based encryption.

I. INTRODUCTION

With the rapid growth of mobile computing and social networking technologies, social network has extended its popularity from the Internet to mobile domain. Personal mobile devices (e.g., smart phones) could communicate with each other for social activities by forming a self-organized multi-hop radio network and maintaining connectivity in a decentralized manner. We call such kind of social networking based on mobile devices that supports instant and pervasive social activities as Pervasive Social Networking (PSN).

Nowadays, Mobile Ad Hoc Network (MANET) has become a practical platform for pervasive social networking and computing, playing as a valuable extension and complement of traditional on-line social networks over the Internet. For example, a user could query people in vicinity using his/her mobile device about which shop is on sale, which movie is recommended to see, or which mobile application should be installed for tagging the locations of photos. The user neighbors can respond these queries by providing their recommendations via PSN. The users could also chat with people nearby for sharing a taxi ride in a flight before landing or affording the cost of a series of movie tickets in front of a movie theatre. Moreover, they can seek services or aids from strangers in vicinity through PSN. People who are strangers but regularly appear in the same public places could want to make an instant appointment for a face-to-face meeting. Particularly, PSN can be applied to collect useful data about an environment in a pervasive and instant manner. This kind of social networking brings extensive social experiences to mobile users, thus is very valuable with unlimited potential, especially when the Internet or cellular networks are temporarily unavailable or costly to access.

Trust plays an important role in PSN for reciprocal activities among nearby strangers. It is a measure derived from direct or indirect knowledge and experiences based on previous interactions and is used to assess the level of belief and dependence put into an entity. Trust helps people overcome perceptions of uncertainty and risk and engages in ‘‘trusted social behaviors’’. During the instant social activities, users are not necessarily acquaintances but more likely to be strangers. Therefore the users need to balance between the benefits received in such reciprocal activities and the risks related to communications with strangers. In this context, it is important to figure out how much users should trust with each other in order to make a social decision about how to disclose and share personal private information.

In order to avoid malicious eavesdropping in PSN, it is crucial to secure PSN communications. It is important to set up a secure communication channel among personally trusted nodes for a serious talk. Since PSN could be established on the basis of a self-organized system and in many scenarios (e.g., disasters and military activities) a centralized server is hard to be connected, a secure communication and access control scheme based on a centralized server becomes inapplicable in the context of PSN scenarios.

We proposed a centralized scheme to control data access based on general trust levels generated by a trusted server and/or local trust levels assessed by each node [3], [30]. But in the case of an urgent disaster and a military activity, this solution becomes impractical due to unavailability of a centralized trusted server. The weakness of such a scheme is the server could be the target of DoS/DDoS attacks or other kinds of attacks. Once the server is broken or crashed down, the reliability of the whole scheme could be terribly influenced. Obviously, one possible solution of this problem is to setup a backup server, but connection availability could still be a question in some practical situations. In practice, a fully trusted server is hard to be established. Internal attackers inside the server could intrude the whole system.

On the other hand, due to the dynamic characteristic of PSN topology and the frequent change of trust relationships in PSN, a message decryption key need to be frequently changed and distributed to each eligible user for securing PSN communications. This introduces heavy communication and computation overheads, which may cause a serious performance bottleneck. How to automatically control data access in a trustworthy and efficient way in PSN is a challenge. Meanwhile, supporting a personalized access control policy for each node is demanded in reality. However, the literature still lacks a secure, efficient and effective scheme to satisfy all expected requirements and solve the issues mentioned above.

This paper proposes two schemes to secure PSN communications purely based on local trust levels evaluated by PSN nodes in a distributed manner. The schemes are designed based on Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [13], [15] and Key-Policy Attribute-Based Encryption (KP-ABE) [14], respectively. We propose using a local trust level to control the access of communication data in PSN based on distributed trust evaluation performed at each PSN node. The PSN node can select other nodes with at least a minimum level of trust to access its communication messages. The nodes with a lower trust level than a pre-defined policy cannot access its communication data. Our work differs substantially from existing work. It applies the trust evaluated by PSN nodes as a specific attribute to control PSN message access in order to secure its communications. Fine-grained and efficient data access control can be implemented and practically applied by mobile users. Specifically, the contribution of this paper can be summarized as below:

1) We motivate securing PSN by controlling its data access based on trust levels in a distributed way. We design two schemes based on CP-ABE and KP-ABE, respectively.

2) To the best of our knowledge, our schemes are ones of the first to secure PSN data communications in a purely distributed manner. They achieve controlling data access based on node trust, which is the key to control data sharing and disclosure.

3) We analyze the security and justify the performance of our proposed schemes through extensive analysis and implementation by comparing the two schemes with each other. In particular, we implement our scheme in both PC and Android platform and develop a prototype system to show its applicability.

The rest of the paper is organized as follows. Section II gives a brief review on related work. Section III introduces the system and threat models and our design goals. Then we provide the detailed description of our schemes in Section IV. Section V provides performance analysis and evaluation, followed by Section VI that introduces a prototype of the proposed schemes to illustrate the applicability and user acceptance of the proposed schemes. Finally, conclusion is presented in the last section.

II. RELATED WORK

There are quite a number of vivid research activities related to social networking and computing. Recent efforts have started to study social communications in the mobile domain.

Several research groups in academia have focused on social activities based on MANET. Stanford Mobile Social Group has developed Junction, a mobile ad hoc and multiparty platform for social applications [5]. Micro-blog [6], developed by SyNRG in Duke University, helps users to post micro-blogs tagged by locations. Ad Social [7], introduced by ETHz Systems Group, provides a pervasive social communication platform. Floating content concept was analyzed based on a theoretical framework to study the fundamental quantities of an ephemeral content sharing service in opportunistic networking [2]. In a proposed floating content system, content is only shared within an anchor zone in a best-effort manner, i.e., copies are kept available within that zone while they are deleted outside the anchor zone [11].

In industry, quite a number of companies, such as Microsoft, Nokia and Intel have conducted researches in the area of PSN. For example, Microsoft Research Asia developed EZ Setup system in order to make a mobile user find services provided by his/her neighbors [8]. The Nokia Instant Community (NIC) developed by the Nokia Research

Center provides an instant social networking platform to allow people in vicinity to communicate, get to know, and share information with each other [9], [12]. Similarly, Intel Berkeley Lab ran a project named Familiar Stranger based on mobile devices to extend feelings and relationships with strangers that people regularly meet but do not interact with each other in a public place [10].

However, trust, security and privacy aspects in PSN have not seriously considered in the above projects. Traditional centralized social networking systems (e.g., Facebook) have not taken user privacy and security into a serious concern. They cannot satisfy instant social networking demands, especially when users do not have the Internet connection, but with location proximity. Issues on trust management for security assurance and privacy enhancement need serious research in order to deploy a successful pervasive social networking system that can be easily accepted by mobile users. A number of crucial issues with regard to trust, security and privacy should be solved. Most existing work did not consider how to control social communication data access based on trust [24], especially in instant and distributed social networking scenarios.

Access control on encrypted data means that only the users with permissions can decrypt encrypted data. The ideal approach is to encrypt each data once, and distribute appropriate keys to users once, so that each user can only decrypt its authorized data. As mentioned already, in PSN, due to the changes of topology and trust relationships, the decryption key should be frequently changed in order to achieve an expected security level. Pure symmetric key based encryption is not good for PSN since the key is hard to be managed in a distributed way. It is complicated to control data access based on trust level and other policies. Public key based encryption is also not suitable for PSN, especially for community based instant social activities. This is because this kind of encryption schemes is not efficient for multicasting and broadcasting data to a group of users. The data owner has to encrypt its data separately for each target receiver. There are also quite a number of schemes to control data access by applying a centralized party in the literature, [18]–[23]. However, they are not feasible to secure PSN communication data since such a party may not exist in PSN.

Attribute-Based Encryption (ABE) [13]–[16] is a new cryptographic technique. In an ABE system, users are identified by a set of attributes rather than an exact identity. Each data is encrypted with an attribute-based access structure, such that only the users whose attributes satisfy the access structure can decrypt the data. ABE has been widely applied into cloud data protection [17], [25]–[27]. Seldom, it is used for securing PSN. Our previous solution relies on a centralized trusted server to generate and issue data access keys [30]. But it is not feasible in some scenarios when the server is not available or trusted. In case the server is hacked, the trustworthiness of the whole system is crashed.

Some existing schemes support access control in a distributed or semi-distributed way. Chatterjee and Das proposed a password-based user access control scheme for hierarchical wireless sensor networks based on ABE [31]. In this scheme, users with an appropriate set of attributes are authorized to access information and resources provided by a cluster header in WSN. However, the structure of this scheme is semi-distributed, a group of users are under the management of the cluster header. Thus, this scheme is also not feasible if the topology of PSN is fully distributed. Cho et al. proposed a distributed composite trust-based public key management scheme for MANETs [32]. It takes the advantage of trust instead of hard security approaches (e.g., encryption or authentication techniques) to improve the performance of key management while achieve security requirements. Users with a trust value higher than a threshold can request relevant public key of others. However, the trust threshold set in this scheme is constant, not a criterion to determine the public key distribution, which are coarse-grained. The authors focused on studying the impact of the trust threshold based public key management on security vulnerability, availability, and communication cost. Bernabe et al. proposed a trust-aware access control mechanism for Internet of Things [33]. In this scheme, trust value of a device is evaluated by taking into account four dimensions (e.g. quality of service, reputation, security aspects and social relationships). The evaluated trust is then used to make authorization decisions by other devices. This scheme improves the flexibility of access control in IoT by applying device trust. However, the confidentiality of device information and key management issues were out of consideration in this work. Rantos et al. proposed a policy based access control scheme in low-power and lossy networks (LLN) in order to protect user resources. Users establish a set of policies to control the access to their resources and can also change and adapt the policies to new environmental parameters. This mechanism makes the scheme highly flexible and robust. However, the confidentiality of user recourses and key management were not well considered. The work presented in this paper focuses on securing PSN communication data based on social trust by applying ABE in a distributed manner.

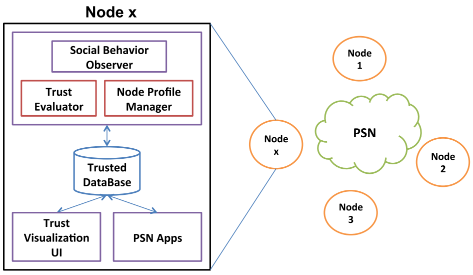


FIGURE 1. **A system model.**

III. PROBLEM STATEMENT

A. SYSTEM AND THREAT MODELS

We consider a PSN system involving only one kind of entities, as illustrated in Fig. 1: the PSN nodes that interact with each other for instant social communications via PSN apps.

As integrity and confidentiality of some instant social communications are crucial, it is important to ensure data security in PSN. We assume that nodes should be able to authenticate with each other. The nodes may not trust with each other. Some nodes may maliciously eavesdrop PSN communication messages to pursue personal benefits. Secure communications among trustworthy nodes in PSN are expected. In addition, each node has a public/secret key pair. The public key (e.g., made by the nodes based on its current identifier) is shared with other nodes if needed for the purpose of authentication and secure communications. Since PSN nodes are mostly strangers, delegation is not allowed among them.

B. DESIGN GOALS

To achieve trustworthy data access in PSN without a trusted server support, our design should achieve the following security and performance goals.

- Security and safety: the communication data in PSN can only be accessed by eligible nodes that are trustworthy enough; the control of data access is conducted according to the trust evaluated by each node based on pervasive social networking performance and experiences;

- Personalization: the scheme allows each node to define its own policies to control its data access;

- Lightweight: the scheme controls PSN data access with light computation and communication overheads.

IV. THE PROPOSED SCHEME

In this section, we design a distributed trust management system in PSN and describe how to embed it into our proposed scheme.

A. DISTRIBUTED TRUST MANAGEMENT SYSTEM IN PSN

Fig. 1 illustrates the structure of a distributed trust management system in PSN. At each node, a User Behavior Observer records node social behaviors. A set of PSN applications provides a user interface and social networking functionality for the node user to conduct social activities in a pervasive way. A Trust Evaluator evaluates the local trust of other nodes based on their social behaviors and provides the results to the user via a Trust Visualization UI. In addition, a Node Profile Manager is used to maintain node personal information and responsible for generating cryptography keys related to securing PSN communications. A Trusted Database stores all data such as the collected social behavior data, trust evaluation results, user profiles, keys, and other data related to the above functional blocks in a secure manner. With regard to concrete trust evaluation algorithms, we can adopt some existing algorithms, as we have studied [1], [4].

B. NOTATIONS, PRELIMINARIES AND DEFINITIONS

1) BILINEAR PAIRING

Let G and G_T be two cyclic multiplicative groups with the same prime order p, that is, $|G| = |G_T| = p$. Let g be a generator of G. Let us have a bilinear mape: $G \times G \rightarrow G_T$, with the following properties:

- Bilinear: for all $u, v \in G$, and $a, b \in Z_p$, $\mathrm{e}\left(u^a, v^b\right) = e(u, v)^{ab}$.
- Non-degenerate: $\mathrm{e}\,(g, g) \neq 1$ for the generator g.
- Computable: there is an efficient algorithm to compute $e\,(u, v)$ for any $u, v \in G$.

Definition: **Local trust level (LT)** is the trust level evaluated by the PSN node according to the information locally accumulated based on node identifiers (which can be a pseudonyms). Herein, we divide trust into discrete levels, e.g., LT_i represents the i-th level of LT, $i \in \left[1, \bar{\mathrm{I}}\right]$, where $\bar{\mathrm{I}}$ is the maximum level of LT.

2) REQUIRED KEYS AND SYSTEM SETUP

During system setup, every node u maintains public key PK_u that is used by other nodes to generate its personalized secret attribute keys, and secret key SK_u, which is used in the decryption operation related to PK_u. Generation of PK_u and SK_u is the task of node u. The keys SK_u and PK_u are bound to the unique identifier of node u, which can be a pseudonym. This binding is crucial for the verification of the trust level of a node.

Each node maintains a secret key SK_u that is used to issue secret attribute keys to other nodes based on a local trust level. It is also used to generate the public key of attribute LT of node $u - PK(LT\_i, u)$. We denote the representation of the attribute of local trust as LT_i. For every attribute with representation (LT_i, u) there is a public key, denoted $PK(LT\_i, u)$, which is generated by node u and is used to encrypt symmetric key $S\_(u)$ for encrypting communication data (i.e., PSN messages) of u, aiming to control access based on the local trust level evaluated by u. The corresponding secret attribute keys for decrypting the cipher-key encrypted by $PK(LT\_i, u)$ are personalized for eligible nodes and issued by node u. To prevent collusion, every node gets a different secret attribute key that only it can use. A secret attribute key of the attribute LT, issued for eligible node u by node u is denoted as $SK(LT\_i, u, u')$. We call the set of secret keys that user u' has (i.e., SK_u' and $SK(LT\_i, u, u')$) as its key ring. Table 1 summarizes the keys used in the proposed schemes.

To effectively secure PSN, we resort to controlling the data access by applying ABE [13]–[16]. ABE is a new cryptographic technique that identifies users by a set of attributes rather than an exact identity. It has developed into two branches: CP-ABE [13], [15] and KP-ABE [14]. In CP-ABE, the users' keys are associated with a set of descriptive attributes and ciphertexts are associated with an access policy. Reversely, in KP-ABE, each ciphertext is labeled with a set of descriptive attributes and each private key is associated with an access structure that specifies which type of ciphertext the key can decrypt.

C. THE SCHEME BASED ON CP-ABE

1) SCHEME DESIGN

The scheme based on CP-ABE consists of a number of fundamental algorithms: **TrustEvaluation**, **InitiateNode**,

TABLE 1. System keys.

Key	Description	Usage
PK_u	The public key of node u;	The unique ID of a node and the key for verification of the node attributes; for evaluation of node local trust and generation of personalized secret attribute key for u;
SK_u	The secret key of node u;	For decryption (to get personalized secret attribute key);
$PK(LT\_i, u)$	The public key of attribute Local Trust generated by u;	For encryption of the symmetric key of node u;
$SK(LT\_i, u, u')$	The secret key of attribute Local Trust for node u' issued by u;	For decryption of the symmetric key of node u;
$S\_(u)$	The symmetric key of node u;	For encryption of PSN communication data of node u;
$S'\_(u)$	The refreshed symmetric key of node u.	For encryption of new PSN communication data of node u.

IssueLocalTrustPK, **IssueLocalTrustSK**, **EncryptKey**, **DecryptKey**, **Encrypt and Decrypt**. The description of the eight algorithms is as follows:

*TrustEvaluation(**u**, **u**′, **PSN_u**′):* The algorithm takes as input the node identities of u and u' (generally the unique node identifier) and the social behavior data PSN_u' of node u'. It outputs the trust level of u' locally evaluated by node u. This process is conducted at node u. For the details of some concrete algorithms of trust evaluation, refer to our previous work [1], [4].

*InitiateNode (**u**):* The algorithm chooses $e : G \times G \to G_T$, with g as a generator of G. It chooses $mk_u \in Z_p$ and outputs $PK\_u = g^{mk_u}$, which is used to issue secret attribute keys to u. This algorithm randomly chooses $P \in G$ and $y \in Z_p$ and generates $SK\_u = g^y \cdot P^{mk_u}$ that is used for the decryption of ciphertext encrypted by PK_u. This process is conducted at node u.

*IssueLocalTrustPK(**PK_u**, **LT**, **SK_u**):* This algorithm is executed by node u whenever the node would like to control the access of its data based on a locally evaluated trust level. It also chooses uniformly and randomly a hash function $H_{SK\_u} : \{0, 1\} \to Z_p$ from the finite family of hash functions and returns $PK(LT\_i, u)$ for each attribute LT_i, which consists of two parts:

$$PK(LT\_i, u) = \langle PK(LT\_i, u)' = g^{H_{SK\_u}(LT\_i)}, PK(LT\_i, u)'' = e(g, g)^{yH_{SK\_u}(LT\_i)} \rangle.$$

*IssueLocalTrustSK(**PK_u**, **LT**, **SK_u**, **PK_u**′):* This algorithm is executed by node u. It checks whether node u' with PK_u' is eligible regarding LT (e.g., the local trust level of u' is equal or above an indicated threshold). If this is the case, this algorithm outputs $SK(LT, u, u')$ for user u'.

$$SK(LT\_i, u, u') = PK\_u'^{H_{SK\_u}(LT\_i)} = g^{mk_{u'} H_{SK\_u}(LT\_i)}.$$

Otherwise, the algorithm outputs NULL.

*EncryptKey(**S_(u)**, **A**, **PK(LT_i, u)**):* This algorithm takes as input $S\_(u)$, access policy A and $PK(LT\_i, u)$ corresponding to the local trust occurring in A. The algorithm encrypts $S\_(u)$ according to A and outputs cipher-key CK. This process is conducted at node u to protect $S\_(u)$. It is executed by node u based on A to protect $S\_(u)$. Policy A is described in Disjunctive Normal Form (DNF) as below:

$$A = \bigvee_{i=j}^{n} (LT\_i),$$

where LT_i denotes the attribute that occurs in the j-th conjunction of A. The encryption key algorithm iterates over all $j = 1, \ldots, n$, generates for each conjunction a random value $R_j \in Z_P$ and constructs CK_j corresponding to each LT_i. The cipher-key CK is obtained as tuple $CK = \langle CK_j, CK(j + 1), \ldots CK_n \rangle$.

$$CK_j = \langle E_j = S\_(u) \cdot PK\ (LT\_j, u)''^{R_j}, \\ E_j' = P^{R_j}, E_j'' = PK(LT\_j, u)'^{R_j} \rangle.$$

For example, in a trust management system, $\bar{\text{I}} = 5$ and one node would like other nodes with local trust level over 4 to access its data. The node encrypts its data with policy $A = (LT\_4) \vee (LT\_5)$, and $CK = \langle CK_4, CK_5 \rangle$.

*DecryptKey(**CK**, **A**, **SK_u**′, **SK(LT_i, u, u**′**)**):* The algorithm takes as input a cipher-key produced by the EncryptKey algorithm, access policy A, under which CK was encrypted, and a key ring $(SK\_u, SK(LT, u, u))$ for node u. It decrypts CK and outputs the corresponding plain-key $S\_(u)$ if the attributes were sufficient to satisfy A. Otherwise it outputs NULL. Concretely,

$$S\_(u) = E_j \cdot \frac{e\left(E_j', SK(LT\_i, u, u')\right)}{e\left(E_j'', SK\_u'\right)}.$$

Let $H_{SK\_u}(LT\_i) = x$, then,

$$\begin{aligned} & E_j \cdot \frac{e\left(E_j', SK\left(LT\_i, u, u'\right)\right)}{e\left(E_j'', SK\_u'\right)} \\ &= S\_(u) \cdot PK\ (LT\_i, u)''^{R_j} \cdot \frac{e\left(P^{R_j}, SK\left(LT\_i, u, u'\right)\right)}{e\left(PK\ (LT\_i, u)'^{R_j}, SK\_u'\right)} \\ &= S\_(u) \cdot e\,(g, g)^{yxR_j} \frac{e\left(P^{R_j}, g^{mku'x}\right)}{e\left(g^{xR_j}, g^y \cdot P^{mku'}\right)} \\ &= S\_(u). \end{aligned}$$

This process is executed when a node receives a PSN message. It firstly checks the encryption policy A, then conducts decryption with the key rings to get the symmetric key. Once getting the plain key and it is still valid, u can use it to decrypt the messages sent from u.

*Encrypt(**S_(u)**, **M**):* The Encrypt algorithm takes as input communication message M and $S\_(u)$. It encrypts M and outputs ciphertext CT. This process is conducted at a node to protect its communication data with $S\_(u)$.

Decrypt($\boldsymbol{S\_(u)}$, $\boldsymbol{CT}$): The Decrypt algorithm takes as input CT and $S\_(u)$. It decrypts CT and outputs M. This process is conducted at node u to gain the content of PSN communication data of node u.

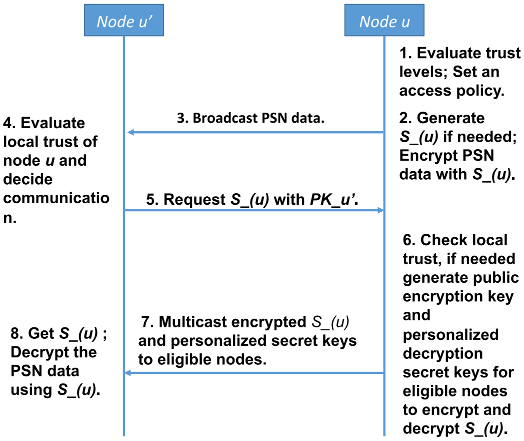

Node u'
Node u
1. Evaluate trust levels; Set an access policy.
3. Broadcast PSN data.
2. Generate S_(u) if needed; Encrypt PSN data with S_(u).
4. Evaluate local trust of node u and decide communicatio n.
5. Request S_(u) with PK_u'.
6. Check local trust, if needed generate public encryption key and personalized decryption secret keys for eligible nodes to encrypt and decrypt S_(u).
7. Multicast encrypted S_(u) and personalized secret keys to eligible nodes.
8. Get S_(u) ; Decrypt the PSN data using S_(u).

FIGURE 2. Control PSN data access based on CP-ABE scheme.

2) PROCEDURE OF CP-ABE BASED SCHEME

We illustrate the procedure of secure pervasive social networking based on the CP-ABE based scheme Fig. 2. If a node wants to secure its communication data based on its local trust evaluation, it generates $S\_(u)$ and corresponding public key and personalized secret keys based on the local trust level for encrypting and decrypting $S\_(u)$. It issues the personalized secret keys to those nodes that satisfy the decryption conditions. The encrypted communication data are broadcast to nearby nodes. Only those nodes that satisfy the access control policy can decrypt $S\_(u)$ and then get the plain data. In case that the local trust levels of some nodes are changed, the node will regenerate a new symmetric encryption key $S'\_(u)$ and issue it to current eligible nodes using ABE. Latest communication data sent from the node will be then encrypted with $S'\_(u)$. The detailed procedure is described as below.

1) Node u evaluates trust level of other nodes by calling TrustEvaluation(u, u', PSN_u') and sets access policy A for its data communications;

2) Node u generates $S\_(u)$ if needed when the trust level of other nodes has changed. It encrypts its data M with $S\_(u)$ by calling Encrypt($S\_(u)$, A, M);

3) Node u broadcasts its encrypted data CT to nearby nodes.

4) Node u' ($u' = 1, 2, \ldots$) gets CT and would like to access after evaluating or checking node u's trust level by calling TrustEvaluation(u', u, PSN_u).

5) If the trust level of node u satisfies u', node u' would first check whether their plain key is still valid. Node u' sends a request with PK_u' for getting $S\_(u)$ from node u if the old $S\_(u)$ is useless.

6) Node u checks the local trust level of node u'. If it satisfies its access control policy and u has not generated the public encryption key and/or corresponding personalized secret attribute decryption keys for the requesting nodes, node u generates proper keys for eligible nodes by calling IssueLocalTrustPK (PK_u, LT, SK_u) and IssueLocalTrustSK(PK_u, LT, SK_u, PK_u'). It encrypts the symmetric key with the public encryption key $PK(LT\_i, u)$ by calling EncryptKey($S\_(u)$, A, $PK(LT\_i, u)$) to get CK.

7) Node u sends (via multicast or unicast) CK and personalized secret decryption keys to eligible nodes in a secure channel (e.g., by applying Public Key Cryptosystem).

8) Eligible node u' decrypts CK with its personalized secret keys by calling DecryptKey(CK, A, SK_u', $SK(LT\_i, u, u')$). Then, u' gets plaintext of CT by calling Decrypt($S\_(u)$, CT).

D. THE SCHEME BASED ON KP-ABE

1) SCHEME DESIGN

The scheme based on KP-ABE also consists of eight fundamental algorithms as described below.

TrustEvaluation($\boldsymbol{u}$, $\boldsymbol{u'}$, $\boldsymbol{PSN\_u'}$): This algorithm is designed the same as before.

InitiateNode($\boldsymbol{u}$): The algorithm chooses $e : G \times G \rightarrow G_T$, with g as a generator of G. As referred above, the trust of each node is divided into a number of discrete levels: $LT = \{LT\_1, LT\_2, \ldots, LT\_\bar{I}$. For each attribute LT_i, the algorithm chooses number t_i uniformly at random in Z_p and also randomly selects y in Z_p.

IssueLocalTrustPK($\boldsymbol{LT\_i}$, $\boldsymbol{u}$): The algorithm is executed by node u when the node would like to encrypt $S\_(u)$ with attribute LT.

$$PK\ (LT, u) = \{G, G_t, g, Y = (g, g)^y, T_{LT\_1} = g^{t_1}, \\ T_{LT\_2} = g^{t_2}, \ldots T_{LT\_n} = g^{t_n}\}.$$

The master key MK is $\{y, t_1, t_2, \ldots, t_n\}$ and node u keeps it as a secret. Obviously, the node can encrypt the data with other attributes apart from trust level, but the trust level is the only attribute in our scheme. This algorithm directly outputs the public attribute key that is used to encrypt $S\_(u)$.

IssueLocalTrustSK($\boldsymbol{LT\_i}$, $\boldsymbol{u}$, $\boldsymbol{u'}$): The algorithm is executed by node u according to the LT of node u' and outputs secret attribute key $SK(LT\_i, u, u')$:

$$SK(LT\_i, u, u') \\ = \{D_1 = g^{q_r(0)/t_1}, D_2 = g^{q_r(0)/t_2}, \ldots, D_i = g^{q_r(0)/t_i}\}.$$

In KP-ABE, $SK(LT\_i, u, u')$ is associated with access structure A, and q_r is a chosen polynomial for node u' in the access structure tree [14]. We issue an access control tree T with root node u'. For node u', we set a threshold value to be 1, choose a polynomial q_r with degree $d_{u'} = 0$, and set $q_r\ (0) = y$. Each leaf node in tree T has an attribute t_i. In our scheme, the access tree has two layers, the root of the tree is an OR gate and the leaves consist of the trust levels that are less than or equal to the LT of node u'.

For example, node u would like to encrypt its data with attributes $LT = 4$. That means the node could access the data once its local trust level evaluated by node u exceeds level 4. The access structure A in the secret attribute key can be written as:

$$A = \bigvee\nolimits_{i=1}^{x} (LT\_i), \quad \text{where } x \text{ is the LT of node } u'.$$

Only the access structure A contains attribute LT_i can make the node decrypt a massage encrypted under LT_i.

EncryptKey($\boldsymbol{S\_(u)}$, $\boldsymbol{LT\_i}$, $\boldsymbol{PK(LT\_i, u)}$): The EncryptKey algorithm takes as input $S\_(u)$ and $PK(LT\_i, u)$ corresponding to the LT_i of node u. The algorithm outputs cipher-key CK.

$$CK_i = \langle LT\_i, E' = S\_(u)Y^S, E = T^s_{LT\_i} \rangle,$$
$$\text{where } s \in Z_p.$$

DecryptKey($\boldsymbol{CK}$, $\boldsymbol{A}$, $\boldsymbol{SK(LT\_i, u, u')}$): This algorithm takes as input CK, A and $SK(LT\_i, u, u')$. It decrypts CK and outputs the corresponding plain-key $S\_(u)$ as below.

$$e\,(D_i, E) = e(g^{q_r(0)/t_i}, g^{s*t_i}) = (e(g, g)^{q_r(0)})^s.$$

If $SK(LT\_i, u, u')$ matches CK, $q_r(0)$ would be y and the above function returns Y^s [5].

$$e\,(D_i, E) = Y^S.$$

Then we can get $S\_(u) = E'/Y^S$.

Otherwise it outputs NULL.

This algorithm is executed when a node receives a message. It conducts decryption with the key rings to get the symmetric key. Once getting the plain key and it is still valid, u' can use it to decrypt the message.

Encrypt($\boldsymbol{S\_(u)}$, $\boldsymbol{M}$): This algorithm is designed the same as in Section 4.3.

Decrypt($\boldsymbol{S\_(u)}$, $\boldsymbol{CT}$): This algorithm is designed the same as in Section 4.3.

2) PROCEDURE OF KP-ABE BASED SCHEME

Fig.3 illustrates the procedure of securing PSN based on the KP-ABE based scheme. There is no big difference between the two schemes. In the scheme based on KP-ABE, node u does not check whether the LT of u' satisfies the requirement of data access, but directly output the corresponding $SK(LT\_i, u, u')$ for node u'. This is the main difference between the two schemes. A node decrypts CK with its personalized secret keys by calling DecryptKey(CK, A, $SK(LT\_i, u, u')$). But the node may not decrypt CK successfully. Because node u generates $SK(LT\_i, u, u')$ based on LT_i evaluated by node u and this $SK(LT\_i, u, u')$ can only decrypt CK encrypted by the attribute LT_j that is equal or less than LT_i.

V. PERFORMANCE ANALYSIS & EVALUATION

In this section, we first discuss the advantages of the designed schemes in a general way. Then, we analyze the their performance from the view of computation complexity, communication cost and scalability. Furthermore, we implement the schemes and evaluate their operating performance based on a number of simulations. Finally, we develop a prototype to further illustrate the applicability and user acceptance of the proposed schemes.

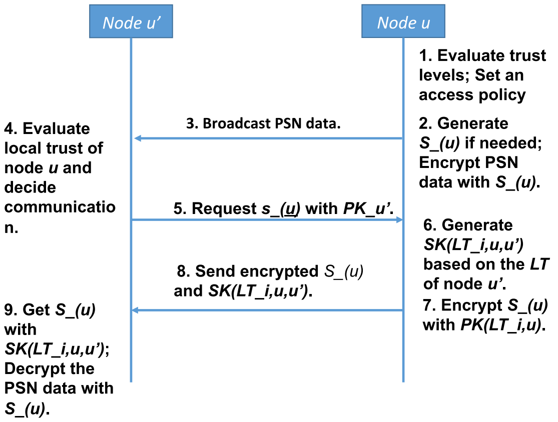

Node u'
Node u
1. Evaluate trust levels; Set an access policy
2. Generate S_(u) if needed; Encrypt PSN data with S_(u).
3. Broadcast PSN data.
4. Evaluate local trust of node u and decide communicatio n.
5. Request s_(u) with PK_u'.
6. Generate SK(LT_i,u,u') based on the LT of node u'.
7. Encrypt S_(u) with PK(LT_i,u).
8. Send encrypted S_(u) and SK(LT_i,u,u').
9. Get S_(u) with SK(LT_i,u,u'); Decrypt the PSN data with S_(u).

FIGURE 3. Control PSN data access based on KP-ABE scheme.

A. DISCUSSIONS ON ADVANTAGES

1) FLEXIBILITY

It is flexible for the proposed schemes to control data access in PSN without a centralized server to manage cryptographic keys. Both proposed schemes support data access control based on distributed trust evaluation at each PSN node. Each node locally manages the corresponding cryptographic keys. No matter new node joining or old node leaving, each node u only generates one key $SK(LT\_i, u, u')$ for another node u' based on trust evaluation. Each node issues a personalized decryption secret key to another node according to the result of its local trust evaluation.

2) PERSONALIZED ACCESS CONTROL

The two schemes support distributed data access control in PSN based on trust management. It supports personal access control policies handled by each individual node. Each node can set its own data access policy and manage keys by itself. Various data access policies can be considered and integrated into local trust evaluation. Thus, it is possible to support complicated access policies through trust evaluation.

3) SECURITY

The security goal of our schemes is to guarantee that only the users whose trust satisfies with the access control policy of the PSN node can access its communication data. The security of the scheme is ensured by the ABE theory and symmetric key encryption theory. The security is further ensured by the fine-grained encryption mechanism controlled by the frequency of trust evaluation at each node. The trust management is distributed, thus can fight against DoS and DDoS attacks on a centralized server [3], [30]. The symmetric key is regenerated if needed when the trust level of some nodes are deduced, which ensures the level of expected security. The detailed security proof about data access control based on trust levels by applying ABE is provided in our previous work [30].

4) DATA CONFIDENTIALITY

In the proposed schemes, the PSN data are encrypted using a symmetric key, and the key is encrypted using the EncryptKey algorithm. Assumed that the symmetric key algorithm is secure, e.g., using a standard algorithm such as AES, the data confidentiality of our proposed scheme merely relies on the security of the ABE algorithm. We have rigorously proved the security of ABE scheme in [30]. Thus, the data confidentiality of the schemes can be ensured.

5) REVOCATION

When a node that knows the symmetric key is revoked due to trust level changes (e.g., going below the pre-defined threshold), the node can re-generate a new symmetric key for later communication data encryption and issue it to currently eligible nodes by applying ABE.

6) EFFICIENCY

In the proposed schemes, the data access control policy is as simple as only related to trust levels. Thus, the computation complexity of encryption and decryption is greatly reduced. Our schemes are scalable to support various access control demands in PSN with trust management support. We reduce the complexity of cryptographic computation by integrating trust evaluation into fine-grained access control. Any complicated attributes that should be considered in the access control policy can be taken into account during the process of trust evaluation.

B. PERFORMANCE ANALYSIS

This section evaluates the performance of our proposed schemes in terms of computation complexity, communication cost, and scalability.

1) COMPUTATION COMPLEXITY

We analyze the computation complexity of the following algorithms: IssueLocalTrustPK, IssueLocalTrustSK, EncryptKey, and DecryptKey.

In the scheme based on CP-ABE, each of the algorithms InitiateNode, IssueLocalTrustPK and IssueLocalTrustSK contains a constant number of exponentiation operations on group G. So the computation complexity of these algorithms is $\mathcal{O}(1)$.

The main computation overhead of encryption operation is the encryption of the message using the symmetric key in the Encrypt algorithm. The complexity of the former depends on the size of the underlying message and is inevitable for any cryptographic method.

The algorithm EncryptKey requires three exponentiation operations on group G for each conjunction of A. So the computation complexity of the encryption operation is $\mathcal{O}(3n)$, where n denotes the number of conjunctions in A, $n \leq \bar{I} + 1$

The only computationally expensive operation presented in DecryptKey is caused by computing exactly two bilinear pairings, no matter how complex the access policy is. This follows from the fact that only one CT_j needs to be decrypted. Thus the computation complexity of key decryption is $\mathcal{O}(1)$.

In the scheme based on KP-ABE, for the algorithms InitiateNode, EncryptKey, and DecryptKey, each contains a constant number of exponentiation operations on group G. So the computation complexity of these algorithms is $\mathcal{O}(1)$.

The algorithms IssueLocalTrustPK, and IssueLocalTrustSK require one exponentiation operation on group G for each conjunction of A. So the computation complexity of these algorithms is $\mathcal{O}(n)$, where n denotes the number of conjunctions in A, $n \leq \bar{I} + 1$.

TABLE 2. Comparison of computation complexity.

Operation	CP-ABE scheme	KP-ABE scheme	Our previous scheme [3, 30]
InitiateNode	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
IssueLocalTrustPK	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$
IssueLocalTrustSK	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$
EncryptKey	$\mathcal{O}(3n)$	$\mathcal{O}(1)$	$\mathcal{O}(3n^2)$
DecryptKey	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Table 2 summarizes the computation complexity of each system operation in our proposed schemes and our previous work [3], [30]. Note that the computation complexity of the scheme in [3] and [30] is more efficient than the schemes in [25] and [26], as we have demonstrated [30]. Obviously, the two schemes proposed in this paper are more efficient than existing work [3], [25], [26], [30]. The main reason is that we control PSN data access based on distributed trust evaluation and simplify the structure of data access policy by only applying the local trust level.

2) COMMUNICATION COST

Cipherkey size is an essential part with regard to communication cost. In the scheme based on CP-ABE, a communication message package frame is $Frame = pseudonym, A, CK, CT$. The size of the frame depends on the size of the underlying communication data, in which CT is unavoidable. Herein, we focus on discussing the size of CT, which is impacted by the proposed schemes. CK is associated with the access structure A that contains a number of conjunctions and each one has only one element (LT). CK is composed of CK_i ($j \leq i \leq n$), each of which has three group elements in G. So the size of CK is linear with respect to the number of CK_i. In our scheme, the size of CK_i is about 404 bytes if the size of $S\_(u)$ is 128 bits. Considering the limited number of levels of LT adopted in practice, the structure of the ciphertext is pretty simple and the size is also reasonable. In the scheme based on KP-ABE, CK is much simpler than that in the scheme based on CP-ABE. The size of CK keeps constant, which is about 270 bytes when the size of $S\_(u)$ is 128 bits. Additionally, we apply the access policy and trust evaluation to assist the decision on key generation and exchange in order to minimize the communication cost in various situations. Based on the above analysis, we can see that from the communication cost point of view, applying the scheme based on KP-ABE is more practical, especially for PSN self-organized by mobile devices.

3) SCALABILITY

The goal of scalability can be achieved by our schemes since the complexity of each operation of our scheme is no longer dependent on the number of nodes in the system, as shown in Table 2. For example, in the scheme based on KP-ABE, except for the key generation algorithms, the complexity of all other operations is $\mathcal{O}(1)$. Even for the EncryptKey in the scheme based on CP-ABE, it requires at most $3(\bar{I}+1)$ exponentiation operations. Therefore, the proposed two schemes can serve as an ideal candidate for securing PSN communications.

C. PERFORMANCE EVALUATION

We implemented the two schemes in C Language using a Pairing Based Cryptography (PBC) library (http://crypto.stanford.edu/pbc/) for the algebraic operations. The implementation used a 160-bit elliptic curve group based on the super singular curve $x^3 + x = y^2$ over a base 512-bit finite field. The experiments were conducted in a workstation with Intel Pentium CPU G630 and 2-GB RAM, running Ubuntu 12.04. In our test machine, the pairings in PBC library can be computed in approximately 4.2 milliseconds (ms).

We estimated the four major operations in our schemes based on CP-ABE and KP-ABE: IssueLocalTrustPK, IssueLocalTrustSK, EncryptKey and DecryptKey.

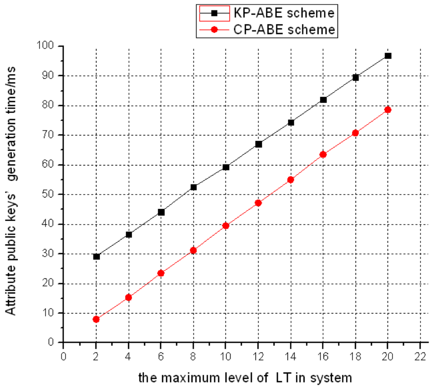


FIGURE 4. Generation time of ABE public keys.

As shown in Fig. 4, the ABE public key generation time is precisely linear with respect to the maximum level of LT in both schemes. We observe that the generation of $PK(LT\_i, u)$ in the scheme based on CP-ABE is more efficient.

In Fig. 5, the ABE secret key generation time is precisely linear with respect to the maximum level of LT in the scheme based on KP-ABE. Because the secret key $SK(LT\_i, u, u')$ is associated with policy A in KP-ABE. But this time is constant in the scheme based on CP-ABE, which does not vary with the maximum level of LT.

Fig. 6 shows the operation time of EncryptKey. We encrypt an AES key (128 bits in our test) with different LT levels.

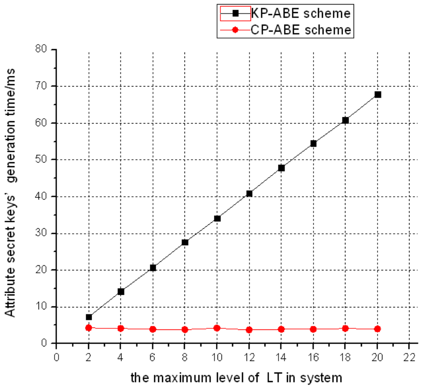


FIGURE 5. Generation time of ABE secret keys.

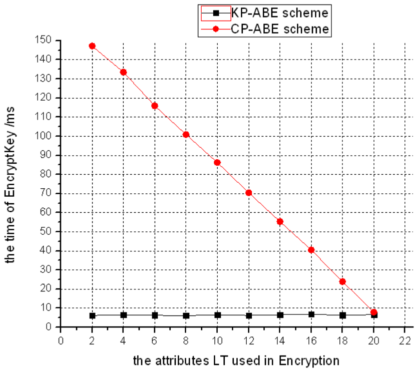


FIGURE 6. Operation time of EncryptKey.

In the KP-ABE based scheme, the AES key would be encrypted by different LT levels, such as $2, 4, \ldots, 20$. The operation time of EncryptKey remains constant, which is about 6 to 7 milliseconds. But it varies with the policy about LT levels in the CP-ABE based scheme, because the AES key is encrypted based on the access structure, such as $LT\_i \geq 2$, $LT\_i \geq 4, \ldots LT\_i \geq 20$. The smaller LT_i threshold used in the access structure, the more CK_i should be generated in this algorithm, thus the operation takes longer time.

Fig. 7 shows the operation time of DecryptKey. We decrypt same CK with different secret keys. The operation time of DecryptKey in both two schemes almost keeps stable. This fact is consistent with our analysis on the computation complexity. Adopting CP-ABE saves the cost on key management while applying KP-ABE can save the computation costs of key encryption and decryption. This fact makes it more feasible in the PSN self-organized by mobile devices since the scheme based on KP-ABE performs better than the scheme based on CP-ABE in terms of communication key encryption and decryption, as well as data encryption and decryption.

Besides, we implemented the scheme based on KP-ABE at the most popular mobile platform Android. The experiments were conducted in a ZTE U9180 mobile phone with Qualcomm Adreno305 (4 core and 1-GB RAM) and Android 4.4 operating system. This is because KP-ABE based scheme is more efficient in data encryption and decryption, thus more suitable for securing the communication data in PSN

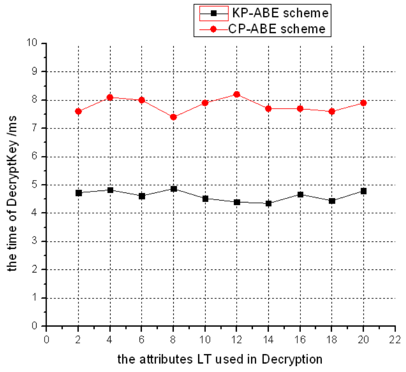


FIGURE 7. Operation time of DecryptKey.

self-organized by mobile devices. The pairings in PBC library can be computed in approximately 38ms in this smart phone. We compared the performance of our KP-ABE scheme in PC Linux platform with Android platform.

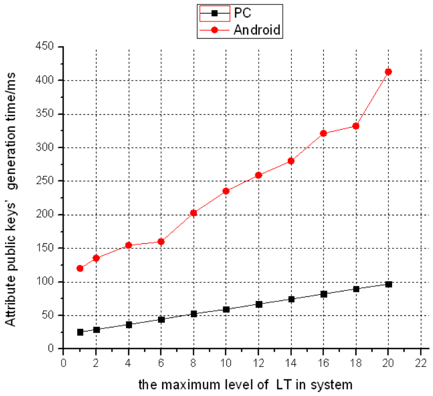


FIGURE 8. Generation time of ABE public keys at PC and Android phone.

As shown in Fig. 8-11, it takes much more time to run each algorithm in the Android phone than the workstation. This result is obvious because the Android phone has much less computational capability than the workstation. But users did not feel the delay in real PSN applications based on the Android phones, as tested and experienced in our demo systems. The operation time of each algorithm in both two platforms keeps the same trend of changes.

VI. DEMO

We implemented a demo mobile app for "secure group chatting" in the Android phones to illustrate the applicability of our schemes. The implementation is based on the scheme based on KP-ABE. The demo illustrates secure group chatting and personal location sharing with privacy preservation in PSN by applying our designed scheme.

A. SECURE GROUP CHATTING

Secure group chatting is popular in our modern life. People prefer to chat with each other in a secure way for some crucial affairs. They need to balance between received benefits

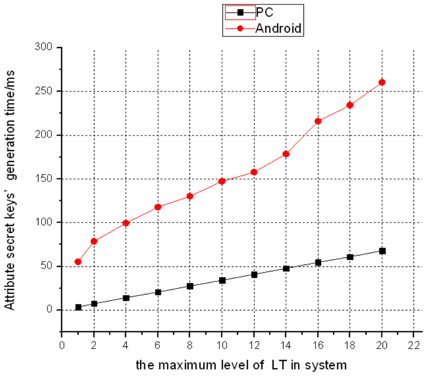


FIGURE 9. Generation time of ABE secret keys in PC and Android phone.

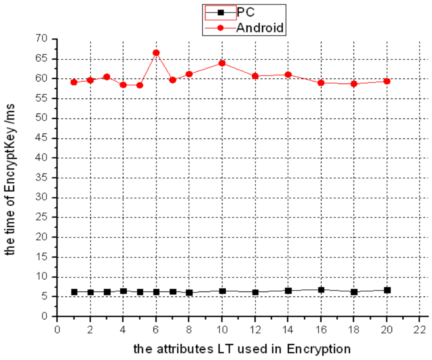


FIGURE 10. Operation time of EncryptKey.

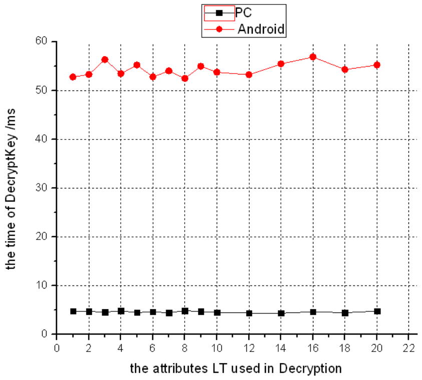


FIGURE 11. Operation time of DecryptKey.

and the risks in communicating with strangers. Our schemes allow group members to chat with each other in a secure way by controlling chatting messages with locally evaluated trust levels.

Fig. 12 shows the User Interface (UI) of secure group chatting. Once the user touches the button of "node initial", the system calls the algorithms InitiateNode and IssueLocalTrustPK at backend before the user joins a group chat in a group chatting room.

The difference of our secure group chatting from the normal group chatting application is users can touch the LT button in UI to set a threshold LT level to control the access structure of encrypted chatting messages. As shown in Fig.13,

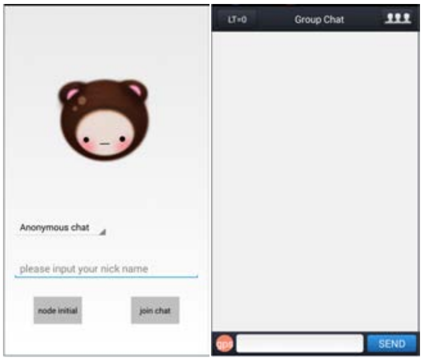
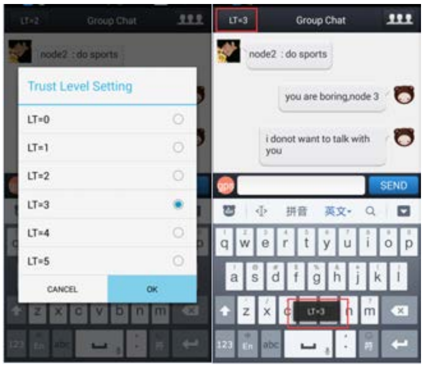


FIGURE 12. User interface of secure group chat.



Trust Level Setting

LT=0 ○
LT=1 ○
LT=2 ○
LT=3 ●
LT=4 ○
LT=5 ○

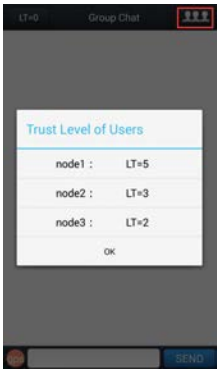
node2 : do sports

you are boring,node 3

i donot want to talk with you

FIGURE 13. User interface of secure group chat.

LT is set as 3. That means the chatting massage is encrypted by $LT = 3$. Only the nodes with trust levels above 3 can access this encrypted message. Obviously selecting Level 0 means sending a plaintext massage.



Trust Level of Users

node1 : LT=5
node2 : LT=3
node3 : LT=2

OK

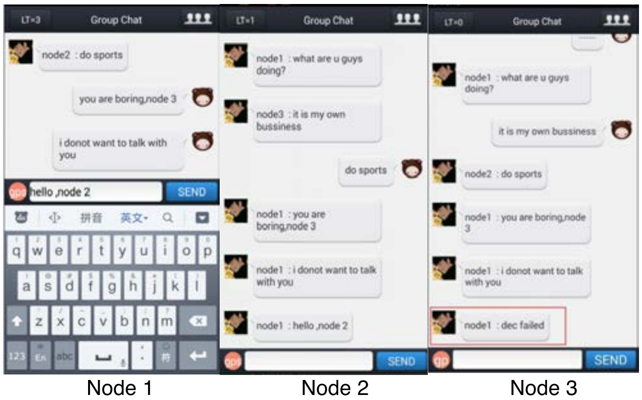
FIGURE 14. Trust level of other nodes evaluated by a local device.

Clicking the button on the top right corner, a user can get the trust level of other nodes evaluated by the local device by running the TrustEvaluation algorithm, as shown in Fig.14. In order to demonstrate the functions of the app, we set a 3 × 3 trust level table in the device database as in Table 3 to demonstrate secure group chatting and location sharing among three nodes.

As shown in Fig.15, firstly, the three nodes can communicate with each other smoothly under the access control policy of $LT \geq 1$. After a while, Node 1 encrypts its massage"hello, node2" with $LT = 3$, then Node 3 cannot decrypt this massage and shows ("dec failed") since its trust level evaluated by node 1 is 2. But Node 2 still can decrypt the massage because its trust level evaluated by node 1 is 3. Every node can still decrypt the massages sent from Node 2 and Node3 since their massage is respectively encrypted by $LT = 1$ and $LT = 0$.

TABLE 3. The trust levels of nodes.

	Node1 trust level	Node2 trust level	Node3 trust level
Node1	$LT = 5$	$LT = 3$	$LT = 2$
Node2	$LT = 2$	$LT = 5$	$LT = 2$
Node3	$LT = 2$	$LT = 2$	$LT = 5$



node2 : do sports

you are boring,node 3

i donot want to talk with you

hello ,node 2

node1 : what are u guys doing?

node3 : it is my own bussiness

do sports

node1 : you are boring,node 3

node1 : i donot want to talk with you

node1 : hello ,node 2

node1 : what are u guys doing?

it is my own bussiness

node2 : do sports

node1 : you are boring,node 3

node1 : i donot want to talk with you

node1 : dec failed

FIGURE 15. Chatting UI of Node 1, 2, & 3.

B. LOCATION PROTECTION

The demo system can help protecting user locations in social networking. We designed a "gps" button on the bottom left corner of the chatting UI. When a user touches this button, a Baidu Map is triggered to show other users' locations if the user satisfies the location access policies of other users, as shown in Fig.16. The red marker indicates the user's own location and the blue ones are other users' locations. Node 1 satisfies all other nodes' location access policies, thus there are three markers in its map. But Node 3 cannot see Node 1's location since its trust level does not satisfy the access policy set by Node 1 ($LT \geq 3$). Therefore, there are only two markers (Node 2's and its own marker) in its map. The GPS location is refreshed every 500 milliseconds. Once a node's trust level is below the threshold set by another node, this node cannot view the GPS information of the another node.

C. USER STUDY

We further studied the user acceptance of our app through a small-scale user study. There were 10 participants (50% female) between 22-29 years old with different backgrounds participating in the experiment. Table 4 shows their basic information, such as major, occupation, time of phone usage

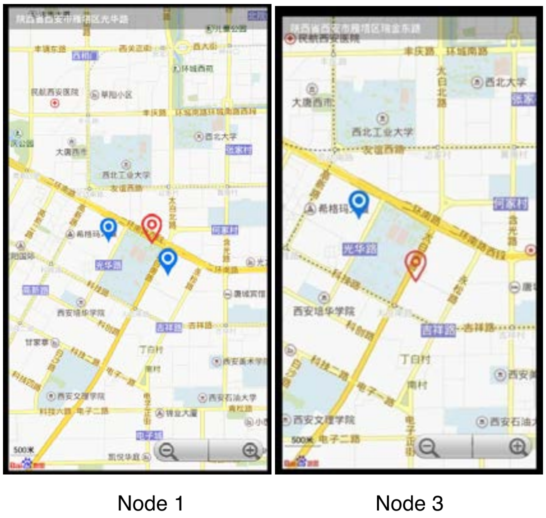


FIGURE 16. GPS locations showed in Node 1 and Node 3.

TABLE 4. Personal information of participants.

User No.	Major	Time of Phone Usage	Personality Type	Occupation
1	Engineering	2-3 hours/day	Phlegmatic	Students
2	Engineering	3-5 hours/day	Sanguine	Students
3	Literature	More than 5 hours/day	Sanguine	Students
4	Math	30 min-1 hour/day	Sanguine	Students
5	Engineering	1-2 hours/day	Phlegmatic	Students
6	Literature	1-3 hours /day	Sanguine	Officer
7	Engineering	Within 1 hour/day	Sanguine	Engineer
8	Engineering	Within 30 min/day	Phlegmatic	Engineer
9	Arts	More than 5 hours/day	Phlegmatic	Website Editor
10	Business	4-5 hours/day	Phlegmatic	Manager

per day and personality type. We kept the diversity of participants background, which implies that their personal feedback collected in this experiment can imply opinions of normal users, especially young people who are main social networking app users.

We installed the secure group chatting app in the Android phones of all participants. They used the app to chat with each other for a couple of hours. After the usage, each participant was asked to fill in a questionnaire as described in Table 5.

The questionnaire was designed based on Technology Acceptance Model (TAM) to study the social acceptance of the app that implements our scheme. TAM indicates that usefulness, ease of use and playfulness lead to user acceptance [28], [29]. This theory also shows that good interface leads to perceived usefulness and ease of use; playfulness causes easy acceptance (attitude). We conducted an interview with the questionnaire to evaluate perceived ease of use, perceived usefulness, interface, playfulness and attitude in terms of the secure group chatting app. The participants were asked to express their agreement on the questionnaire items in Table 5 by applying a 5-point Likert scale: highly disagree (1), disagree (2), neutral (3), agree (4), and highly agree (5).

The interview result is shown in Figure 17 with Standard Deviation (SD). We can see that the secure group chatting app achieved satisfactory evaluation scores with regard to

TABLE 5. Interview questionnaire.

Purpose	Items
Perceived Ease of Use (PEU)	Q1: It was easy for me to use the secure group chatting app.
	Q2: I did not feel any delay caused by the security scheme embedded in the app.
	Q3: The secure group chatting app did not influence my usage of other mobile apps.
Perceived Usefulness (PU)	Q4: The secure group chatting app can protect important information as I expected in group chatting.
	Q5: The secure group chatting app can protect my chatting massages from being received by distrusted group members .
	Q6: The secure group chatting app can provide me the trust information of group members during chatting, which can help me choose a suitable trust level to protect my chatting massages.
Interface (I)	Q7: The UI of the secure group chatting app was simple and easy to follow.
	Q8: The secure group chatting app had a good UI design.
	Q9: The secure group chatting app provided me a good UI to control the access of my chatting messages and my location.
Playfulness (P)	Q10: The way of encryption used in the secure group chatting app was sound.
	Q11: The secure group chatting app was an exciting application.
	Q12: The secure group chatting app provided an interesting way that changes my opinion on secure social chatting.
Attitude (A)	Q13: I would like to use the secure group chatting app.
	Q14: I would like to recommend this app to other people.
	Q15: The secure group chatting app made me feel secure when I used it.

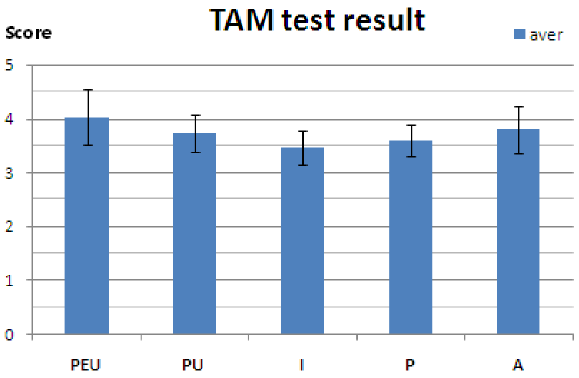


FIGURE 17. TAM test result.

perceived ease of use (Q1 to Q3, $AVE = 4.0$, $SD = 0.51$), perceived usefulness (Q4 to Q6, $AVE = 3.7$, $SD = 0.34$), user interface (Q7 to Q9, $AVE = 3.4$, $SD = 0.32$), playfulness (Q10 to Q12, $AVE = 3.6$, $SD = 0.30$), and attitude (Q13 to Q15, $AVE = 3.8$, $SD = 0.44$). Each testing item gained a high average score (AVE). In particular, perceived ease of use, perceived usefulness and attitude got AVE scores at 4.0, 3.7, and 3.8, which implies that our app is a useful application preferred by the participants. The test also indicated that the UI of the demo should be further improved. Based on the TAM, we can see that the secure group chatting app developed based on our scheme was welcome by the participants due to good user experiences.

VII. CONCLUSION

In this paper, we introduced two schemes to control PSN data access based on the trust evaluated by each PSN node in the situation that no centralized servers are available for key management and data access control. The proposed schemes seamlessly incorporate a PSN trust management framework for securing PSN by applying the ABE theory. Our schemes can flexibly support controlling PSN data access based on local trust with low communication and computation costs. The PSN communication data can be automatically secured since the related cryptographic keys can be automatically generated, issued and managed based on trust evaluation results. We analyzed the performance of the proposed schemes. Comparison with our previous work and simulations based on implementation further show that our schemes are highly efficient. The scheme based on KP-ABE is more feasible for practical usage due to less message encryption time and less communication cost than the scheme based on CP-ABE. At last, we developed a secure group chatting app to further demonstrate the applicability and social acceptance of our schemes.

REFERENCES

[1] Z. Yan, Y. Chen, and Y. Shen, "A practical reputation system for pervasive social chatting," *J. Comput. Syst. Sci.*, vol. 79, no. 5, pp. 556–572, 2013.

[2] J. Ott, E. Hyytiä, P. E. Lassila, J. Kangasharju, and S. Santra, "Floating content for probabilistic information sharing," *Pervas. Mobile Comput.*, vol. 7, no. 6, pp. 671–689, 2011.

[3] Z. Yan, M. Wang, V. Niemi, and R. Kantola, "Secure pervasive social networking based on multi-dimensional trust levels," in *Proc. IEEE CNS*, Washington, DC, USA, Oct. 2013, pp. 100–108.

[4] Z. Yan, Y. Chen, and Y. Shen, "PerContRep: A practical reputation system for pervasive content services," *J. Supercomput.*, vol. 70, no. 3, pp. 1051–1074, 2014.

[5] Stanford MobiSocial Group. *Junction*, accessed on Jan. 5, 2017. [Online]. Available: https://mobisocial.stanford.edu/?page=junction

[6] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt, "Micro-Blog: Sharing and querying content through mobile phones and social participation," in *Proc. ACM MobiSys*, 2008, pp. 174–186.

[7] E. Sarigöl, O. Riva, P. Studei, and G. Alonso, "Enabling social networking in ad hoc networks of mobile phones," *Proc. VLDB Endow.*, vol. 9, no. 2, pp. 1634–1637, 2009.

[8] *EZSetup*, accessed on Jan. 5, 2012. [Online]. Available: http://research.microsoft.com/en-us/groups/wn/mssn.aspx

[9] A. Ahtiainen *et al.*, "Awareness networking in wireless environments: Means of exchanging information," *IEEE Veh. Technol. Mag.*, vol. 4, no. 3, pp. 48–54, 2009.

[10] Intel Berkeley Lab. *Familiar Stranger*, accessed on Jan. 5, 2017. [Online]. Available: http://www.paulos.net/research/intel/familiarstranger/index.html

[11] E. Hyytiä, J. Virtamo, P. Lassila, J. Kangasharju, and J. Ott, "When does content float? Characterizing availability of anchored information in opportunistic content sharing," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 3137–3145.

[12] A. Ahtiainen *et al.*, "Awareness networking in wireless environments," *IEEE Veh. Technol. Mag.*, vol. 4, no. 3, pp. 48–54, Sep. 2009.

[13] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 321–334.

[14] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[15] S. Müller, S. Katzenbeisser, and C. Eckert, "Distributed attribute-based encryption," in *Proc. 11th Annu. Int. Conf. Inf. Secur. Cryptol.*, 2008, pp. 20–36.

[16] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. 24th Int. Conf. Theory Appl. Cryptograph. Techn.*, 2005, pp. 457–473.

[17] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proc. 17th ACM Conf. Comput. Commun. Secur. (Poster)*, 2010, pp. 735–737.

[18] M. Zhou, Y. Mu, W. Susilo, and J. Yan, "Piracy-preserved access control for cloud computing," in *Proc. TrustCom*, 2011, pp. 83–90.

[19] R. Chow *et al.*, "Controlling data in the cloud: Outsourcing computation without outsourcing control," in *Proc. ACM Workshop Cloud Comput. Secur. (CCS)*, 2009, pp. 85–90.

[20] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Proc. Int. Conf. Financial Cryptogr. Data Secur. (FC)*, 2010, pp. 136–149.

[21] Q. Liu, C. Tan, J. Wu, and G. Wang, "Efficient information retrieval for ranked queries in cost-effective cloud environments," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Mar. 2012, pp. 2581–2585.

[22] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in *Proc. USENIX Conf. File Storage Technol. (FAST)*, 2003, pp. 29–42.

[23] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing remote untrusted storage," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2003, pp. 131–145.

[24] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute based systems," *J. Comput. Secur.*, vol. 18, no. 5, pp. 799–837, 2010.

[25] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Mar. 2010, pp. 534–542.

[26] Z. Wan, J. Liu, and R. H. Deng, "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 743–754, Apr. 2012.

[27] G. Wang, Q. Liu, J. Wu, and M. Guo, "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers," *Comput. Secur.*, vol. 30, no. 5, pp. 320–331, 2011.

[28] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quart.*, vol. 13, no. 3, pp. 319–340, 1989.

[29] V. Venkatesh and H. Bala, "Technology acceptance model 3 and a research agenda on interventions," *Decision Sci.*, vol. 39, no. 2, pp. 273–315, 2008.

[30] Z. Yan and M. Wang, "Protect pervasive social networking based on two-dimensional trust levels," *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–12, 2014, doi: 10.1109/JSYST.2014.2347259.

[31] S. Chatterjee and A. K. Das, "An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks," *Secur. Commun. Netw.*, vol. 8, no. 9, pp. 1752–1771, 2015.

[32] J.-H. Cho, I.-R. Chen, and K. S. Chan, "Trust threshold based public key management in mobile ad hoc networks," *Ad Hoc Netw.*, vol. 44, pp. 58–75, Jul. 2016

[33] J. B. Bernabe, J. L. H. Ramos, and A. F. S. Gomez, "TACIoT: Multidimensional trust-aware access control system for the Internet of Things," *Soft Comput.*, vol. 20, no. 5, pp. 1763–1779, 2016.

[34] K. Rantos, K. Fysarakis, C. Manifavas, and I. G. Askoxylakis, "Policy-controlled authenticated access to LLN-connected healthcare resources," *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–11, 2015, doi: 10.1109/JSYST.2015.2450313.

CHAOYIN HUANG received the B.Sc. degree in electronic information science and technology and the master degree in information security from Xidian University, Xi'an, China, in 2011and 2016, respectively.

ZHENG YAN (M'06–SM'14) received the B.Eng. degree in electrical engineering and the M.Eng. degree in computer science and engineering from Xi'an Jiaotong University, Xi'an, China, in 1994 and 1997, respectively, the M.Eng. degree in information security from the National University of Singapore, Singapore, in 2000, and the Lic.es Sci. in electrical engineering and the D.Sc. degree in technology from the Helsinki University of Technology, Helsinki, Finland, in 2005 and 2007, respectively. She is currently a Professor with Xidian University, Xi'an, and a Visiting Professor with Aalto University, Espoo, Finland. She authored over 150 peer-reviewed publications and solely authored two books. She is the inventor and co-inventor of 50 patents and PCT patent applications. Her research interests include trust, security and privacy, social networking, cloud computing, networking systems, and data mining. She served as a Steering, Organization, and Program Committee Member for over 70 international conferences. She serves as an Associate Editor of *Information Sciences*, the IEEE Internet of Things Journal, the IEEE Access Journal, and *Security and Communication Networks*. She is a leading Guest Editor of many reputable journals, including *ACM TOMM*, *Information Fusion*, *FGCS*, the IEEE Systems Journal, *JNCA*, and *MONET*.

NING LI received the B.Sc. degree in communication engineering from the Nanjing Institute of Technology, Nanjing, China, in 2012, and the master's degree in information security from Xidian University, Xi'an, China, in 2016.

MINGJUN WANG received the B.Sc. degree in communication and information systems from Henan Normal University, Xinxiang, China, in 2011. He is currently pursuing the Ph.D. degree in information security with the Xidian University, Xi'an, China. His research interests include security, privacy and trust management in social networking, 5G and cloud computing.